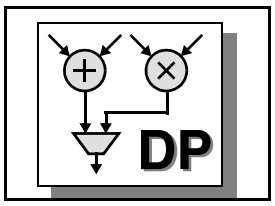

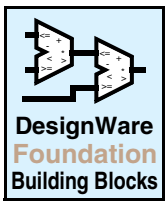

DWBB Datapath Functions Overview

Introduction

The Datapath Functions are HDL functions that can be called in design RTL code (Verilog or VHDL) to implement commonly used higher-level datapath functionality.

How Do the Datapath Functions Work?

The Datapath Functions are written in HDL. They describe some dedicated datapath functionality in synthesizable RTL code. The functions are made available as follows:

- In VHDL through packages
- In Verilog through include files

During design elaboration, the function's HDL code is elaborated together with the design HDL code. During compile, the elaborated design goes through regular synthesis. The sub-design described by the function is automatically optimized by datapath synthesis, together with surrounding datapath functionality.

Benefits of the Datapath Functions

Because the Datapath Functions are written in HDL code and go through the regular synthesis flow, synthesis results are not different than directly describing the same functionality in the HDL code of the design. However, the Datapath Functions offer the following benefits:

- **Ease-of-use:** The desired functionality can be included in the design with a simple function call instead of being written out in HDL code. This alleviates the sometimes cumbersome task of writing correct and synthesis-efficient datapath code.
- **Functional correctness:** The Datapath Functions are pre-verified and therefore guarantee functional correctness.
- **Best QoR:** The Datapath Functions guarantee best possible QoR because their code is optimized for datapath synthesis with Design Compiler.
- **Design integration:** The Datapath Function HDL code tightly integrates into surrounding datapath functionality. This allows high-level optimizations and datapath synthesis to work on larger design partitions, which helps improve QoR even further.
- **Flow integration:** The Datapath Functions are tightly integrated into the Design Compiler datapath synthesis flow. As a result, datapath optimization techniques, such as carry-save representations, can be fully exploited.

Differences to DesignWare Building Block IP

The DesignWare Building Block IP library has the following characteristics:

- Provides implementations for commonly used building blocks.
- Describes dedicated circuit structures that synthesis otherwise cannot generate.
- Complements synthesis.
- Building blocks are synthesized individually, not taking into account the broader design context.

In comparison, the Datapath Functions library has the following characteristics:

- Provides easy-to-use code templates for commonly used datapath functionality.
- Describes higher-level functionality, not low-level circuit structures.
- Relies on datapath synthesis to implement the functionality efficiently.
- Functions are synthesized and optimized in a broader design context, together with surrounding datapath functionality.

Datapath Functions Organization

The Datapath Functions are organized in VHDL packages and Verilog include files. Two versions exist for each:

- Synthesis packages and include files are used for synthesis with Design Compiler. The code is optimized for best possible QoR. The source files are encrypted for IP protection.
- Simulation packages and include files are used for simulation, verification and third-party tools. The code is not optimized for QoR and is not intended for synthesis. The source files are not encrypted and therefore can be used as concise documentation of the functions behavior.

VHDL

The Datapath Functions for VHDL are made available through VHDL packages. Table 1-1 on page 3 summarizes the names of the packages and the corresponding source file names.

- Source files are relative to the installation path $SYNOPSYS/packages/dware/src/.
- Analyzed synthesis source files are in $SYNOPSYS/packages/dware/lib/.

A different functions package needs to be used depending on the employed numeric package (ieee.numeric_std and ieee.std_logic_arith).

Table 1-1 VHDL Packages

Package / File Name	Numeric Package	Description
DW_dp_functions	ieee.numeric_std	Functions package
DW_dp_functions_arith	ieee.std_logic_arith	Functions package
DW_dp_functions_syn.vhd.e	ieee.numeric_std	Encrypted synthesis package file
DW_dp_functions_arith_syn.vhd.e	ieee.std_logic_arith	Encrypted synthesis package file
DW_dp_functions_sim.vhd	ieee.numeric_std	Simulation package file
DW_dp_functions_arith_sim.vhd	ieee.std_logic_arith	Simulation package file

No special setup is required for synthesis with Design Compiler (packages are automatically found). For simulation and third-party tools, the installation path has to be specified and the source files pre-analyzed accordingly.

Verilog

The Datapath Functions for Verilog are made available through include files:

- Include files are named DW_dp_<name>_function.inc. For example, DW_dp_absval_function.inc.
- Encrypted include files for synthesis are in $SYNOPSYS/dw/syn_ver/.
- Include files for simulation are in $SYNOPSYS/dw/sim_ver/.

No special setup is required for synthesis with Design Compiler (include files are automatically found). For simulation and third-party tools, the installation path has to be specified accordingly. For verification with Formality, add `$SYNOPSYS/dw/syn_ver` to the search path.

Usage of Datapath Functions

VHDL

The Datapath Functions are used in VHDL as follows:

- Declare the use of the DW_dp_functions package.
- Call the function in the HDL code.

Example:

```
library IEEE, DWARE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
use DWARE.DW_dp_functions.all;
-- DWARE.DW_dp_functions_arith package if IEEE.std_logic_arith is used

       entity DWF_dp_absval_test is
```

```
  port (a, b, c : in  signed(7 downto 0);
        z       : out signed(15 downto 0));
end DWF_dp_absval_test;

architecture rtl of DWF_dp_absval_test is
begin
  z <= DWF_dp_absval (a * b) + c;
end rtl;
```

Verilog

The Datapath Functions are used in Verilog as follows:

- Define the function parameters to be passed to the function.
- Include the function include file (“DW_dp_<name>_function.inc”).
- Call the function in the HDL code.

Example:

```
module DWF_dp_absval_test (a, b, c, z);

  input  signed  [7:0] a, b, c;
  output signed [15:0] z;

  // Passes the parameter to the function
  parameter width = 16;

  // add "$SYNOPSYS/dw/sim_ver" to the search path for simulation
  `include "DW_dp_absval_function.inc"

  assign z = DWF_dp_absval (a * b) + c;

endmodule
```

Notes

- The function arguments can be arithmetic expressions themselves:

  ```
  assign z = DWF_dp_absval (a * b);
  ```

- The function call can be part of an arithmetic expression:

  ```
  assign z = DWF_dp_absval (a) + c;
  ```

- However, be careful when using functions and arithmetic expressions within the same statement to avoid unintended behavior, due to ambiguities between unsigned and signed types and expression widths. Use intermediate signals if necessary.

  ```
  assign t = a * b;
  assign z = DWF_dp_absval (t) + c;
  ```

  For more information, refer to Coding Guidelines for Datapath Synthesis.

- In any case (function and expression in same statement or use of intermediate signals), the function's functionality and the surrounding datapath functionality are extracted and optimized together as one large datapath partition. Typical datapath optimization techniques (such as carry-save representations) are used wherever possible and beneficial.
- Synthesis results of the Datapath Functions are best if used in conjunction with DC Ultra, which provides the datapath optimizations referred to in this text.

Available Functions

Table 1-2 summarizes the currently available Datapath Functions.

Table 1-2 Datapath Functions

Function Name	Description
DWF_dp_absval DWF_dp_sub_abs	Absolute Value Subtract and Absolute Value
DWF_dp_blend DWF_dp_blend_exact DWF_dp_blend_exact2 DWF_dp_blend2 DWF_dp_blend2_exact2	Graphics Alpha Blend
DWF_dp_count_ones	Counts ones in the argument
DWF_dp_rnd DWF_dp_rndsat	Arithmetic Rounding
DWF_dp_sat DWF_dp_sati	Arithmetic Saturation
DWF_dp_mult_comb DWF_dp_mult_comb_ovfldet DWF_dp_mult_comb_sat DWF_dp_mult_ovfldet DWF_dp_mult_sat	Multiply Functions - Combined Unsigned/Signed Combined Unsigned/Signed with Overflow Detection Combined Unsigned/Signed with Saturation Multiply and Overflow Detection Multiply and Saturate
DWF_dp_sign_select	Sign Selection / Conditional Two's Complement
DWF_dp_simd_add DWF_dp_simd_addc DWF_dp_simd_mult	SIMD operations

Copyright Notice and Proprietary Information

© 2022 Synopsys, Inc. All rights reserved. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com